

WiFi CSI based Liquid Temperature Prediction: A Physics-Guided Machine Learning Approach

Nafeez Fahad, and Eyuphan Bulut

Department of Computer Science, Virginia Commonwealth University
401 West Main St. Richmond, VA 23284, USA
{fahadn, ebulut}@vcu.edu

Abstract—Sensing the temperature of liquids in containers is a critical process in various applications such as food safety, healthcare and environmental monitoring. Traditional approaches usually require a physical contact with the liquid, which may not be suitable for the liquids in sealed containers, and may also pose a risk for contamination in particular for the products used in healthcare. The contactless temperature sensing approaches that mainly rely on optical or laser based solutions, however, can only detect the temperature of a certain point or only the surface temperature of the liquid. Sensing through thermal cameras can provide scalability, however, they come with high costs. In this study, leveraging the ubiquitous availability of low-cost devices integrated with WiFi interfaces, we explore the feasibility of measuring the liquid temperature using fine-grained WiFi features. The proposed approach utilizes the amplitude variations across WiFi channel subcarriers extracted from Channel State Information (CSI) to detect the liquid temperature. In order to enhance the prediction accuracy, we also integrate principles from Newton’s Law of Cooling for regularization during the training of the neural network and propose a physics-guided machine learning framework (PMLF). Through our experiments with different liquids, temperature ranges and containers, we have demonstrated that the proposed cost-effective and scalable solution provides promising results in predicting the liquid temperatures.

Index Terms—Non-contact thermometry, WiFi signals, CSI, liquids.

I. INTRODUCTION

The ubiquity of WiFi devices and signals, along with recent tools [1]–[4] that enable access to the fine-grained WiFi features such as Channel State Information (CSI), has provided a tremendous low-cost and scalable sensing opportunity. By leveraging the signal propagation characteristics embedded in CSI data, researchers have demonstrated the ability to identify individuals [5], track their movements [6], and even distinguish them from pets [7]. Moreover, other studies have utilized this rich data to recognize materials [8] and liquid types [9], and to support indoor security applications [10]–[12].

In this paper, we leverage this CSI-based sensing method for another significant application. Measuring the temperature of liquids in containers without physical contact can have a vital importance in various domains, as shown in Fig. 1. For example, for the fluids used in healthcare, contact-based temperature measuring devices can pose a



Fig. 1: Sensing the temperature of the liquids using wireless signals can provide significant benefits in various domains such as food safety, healthcare, and smart homes.

potential risk of contamination, requiring a non-contact solution. Similarly, non-contact thermometry can allow real-time temperature monitoring of beverages (e.g., milk, water, juice) during storage and transportation without opening containers which can help ensure the safety and compliance with standards. It can also provide an opportunity to sense the temperature of liquids stored in large containers (e.g., water tanks). Finally, by leveraging the WiFi signals available in most of the houses, this can also add another capability to smart home setups.

Existing non-contact temperature sensing methods include optical or laser-based solutions or thermal cameras [13]. These approaches detect the radiation emitted from the object to recognize the associated temperature. One of the issues with these solutions is that they can only measure the temperature of the surface of an object. However, the true temperature of a fluid within a pipe or container can be different. Moreover, the high cost of these solutions can hinder their scalable usage. On the other hand, there are also acoustic or ultrasonic sensing solutions [14], [15]. However, these methods come with limitations. For example, in [15], the microphones of a smartphone are used to estimate the ambient temperature from the sound propagation speed. Since a single device is used to transmit and receive the acoustic signals, the temperature only around the device can be measured. RFIDs are also used for wireless temperature sensing [16], [17]. While they can help reach long ranges and have low-cost, they can only measure temperature at a single location.

Different from these studies, we propose using WiFi CSI data and its fingerprints associated with the liquid medium at different temperature states to develop a novel non-contact thermometry solution. The main contributions of this study

are the following:

- We explore the feasibility of measuring the temperature of a liquid in a container through the utilization of fine-grained WiFi signal information, i.e., CSI. To the best of our knowledge, this is the first work that uses WiFi signals to measure liquid temperature.
- We present a hybrid regression-based approach that leverages both WiFi CSI data and Newton’s Law of Cooling to extract physical features characterizing temporal temperature changes.
- We conduct experiments across various settings and obtain promising results, demonstrating the potential of this method for non-contact liquid temperature measurement.

The rest of the paper is structured as follows. In Section II, we provide an overview of related studies in the literature. In Section III, we discuss some preliminary background information related to the proposed solution, followed by the details of the proposed WiFi CSI-based liquid temperature sensing approach in Section IV. We then provide our experimental evaluation in Section V. Finally, we conclude the study and discuss future directions in Section VI.

II. RELATED WORK

WiFi CSI-based sensing has seen growing interest with applications ranging from human activity and gesture recognition [6], to occupancy monitoring [18], [19], and interesting applications such as fruit ripeness detection [20], and rehabilitation tracking [21], [22].

Sensing environmental features such as air humidity [23], textile wetness [24], soil moisture [25], and the surrounding temperature [26] has also been studied, leveraging the effects on CSI amplitude and phase information in changing environmental settings. This is achieved through various machine learning approaches. For example, [23] uses a KNN classifier to detect humidity in the air, while [24] uses a linear SVM classifier to sense the wetness level in a towel located in a room. In [25], a separate DNN classifier is trained for sensing the soil moisture and texture. In [27], sensing the moisture amounts in wheat crops is explored using an SVM-based classification approach.

The closest work to this study is presented in [26], where the correlation between the temperature of the air and the amplitude of signals is explored. However, no prediction model has been developed or presented in that study. In a related study [28], the detection of indoor fires is studied based on the observation that when the fire starts, the flames affect the transmission of WiFi signals due to additional scattering. A random forest based classifier is used there, producing high accuracy in the detection of fires. Regarding liquids, there are some other studies that aim to detect liquid types [29], or liquid level in a container [9]. However, to the best of our knowledge, there is no other study that leverages WiFi CSI data for measuring the liquid temperature.

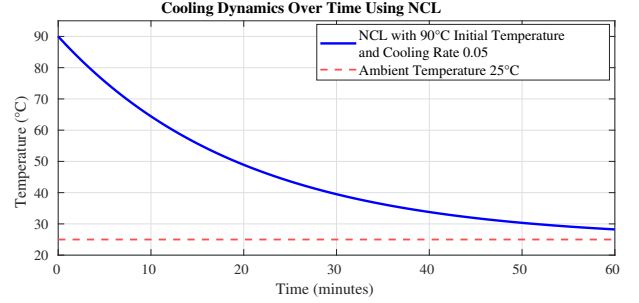


Fig. 2: Temperature evolution over time based on Newton’s Cooling Law

Besides WiFi sensing, other techniques are also used for non-contact thermometry solutions. These include optical or laser based solutions [13], acoustic or ultrasonic sensing solutions [14], [15] and RFID based solutions [16], [17]. These solutions can either sense the temperature in a specific point or have high costs. The proposed WiFi CSI based solution can alternatively provide a low-cost and potentially a scalable solution.

III. PRELIMINARIES

A. Newton’s Cooling Law

Newton’s Cooling Law (NCL) describes how an object exchanges heat with its surrounding environment, assuming the ambient temperature remains constant. The law is expressed by the heat exchange rate defined as,

$$\frac{dT(t)}{dt} = -k(T(t) - T_{\text{env}}), \quad (1)$$

where $T(t)$ is the temperature of the object (e.g., liquid in our case) at time t , T_{env} is the ambient (environmental) temperature, $k > 0$ is the cooling constant (which will depend on the properties of the liquid and the container), characterizing the rate of heat dissipation. The solution of this differential equation gives the temperature value at a given time t by [30]:

$$T(t) = T_{\text{env}} + (T_0 - T_{\text{env}})e^{-kt}, \quad (2)$$

where T_0 is the initial temperature of the liquid and e^{-kt} term represents the exponential decay of the temperature difference over time.

The exponential decay e^{-kt} implies that the temperature difference $|T(t) - T_{\text{env}}|$ decreases at a rate proportional to the current difference. Over time, $T(t)$ asymptotically approaches T_{env} representing thermal equilibrium.

The cooling constant k encapsulates the system’s thermodynamic properties, including the medium’s thermal conductivity, the object’s surface area-to-volume ratio, and the heat transfer coefficient. This makes k a critical parameter in characterizing the cooling behavior of liquids or solids.

Fig. 2 illustrates the temperature evolution over time as described by Eq. 2. Starting from an initial temperature T_0 , the object’s temperature decreases exponentially as it

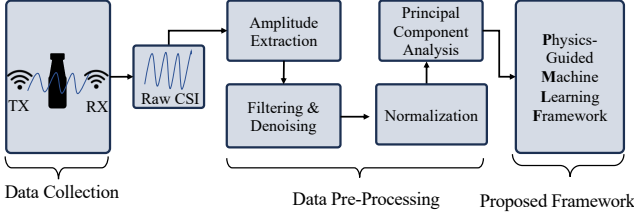


Fig. 3: System Overview

approaches the ambient temperature T_{env} . The red dashed line emphasizes the equilibrium position, at which the temperature variation is minimal. This phenomenon highlights the importance of the exponential decay factor e^{-kt} in defining the cooling process.

In this study, *NCL* provides the theoretical basis for modeling temperature changes in liquids during the cooling process. As the liquid's temperature drops, it follows an exponential decay pattern, which we describe with Eq. 2. At the same time, the CSI from WiFi signals interacting with the liquid also changes in a way that correlates with these temperature shifts.

By applying the principles of *NCL*, we can refine the relationship between the liquid's temperature and the variations in CSI data. Key factors to consider are the cooling constant k and the temperature at a given time $T(t)$, as they help link the physical cooling process to the behavior of the signals observed.

This theoretical foundation enables the integration of temperature dynamics into WiFi sensing, facilitating the use of CSI for non-contact temperature estimation.

B. WiFi Channel State Information

Channel state information provides insight into how the signal propagates between a transmitter (TX) and a receiver (RX). This includes the reflections and scattering from the environment and surrounding objects over multiple paths. The channel is usually described using,

$$\mathbf{y} = \mathcal{H}\mathbf{x} + \eta,$$

where $\mathbf{y}$ and $\mathbf{x}$ represent the received and transmitted signals, respectively; η denotes the noise in the channel; and $\mathcal{H}$ denotes the CSI matrix. $\mathcal{H}[t]$ matrix over a time frame t can be represented by

$$\mathcal{H}[t] = \begin{bmatrix} \mathbf{h}_1[1] & \mathbf{h}_1[2] & \dots & \mathbf{h}_1[t] \\ \mathbf{h}_2[1] & \mathbf{h}_2[2] & \dots & \mathbf{h}_2[t] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_S[1] & \mathbf{h}_S[2] & \dots & \mathbf{h}_S[t] \end{bmatrix},$$

where S denotes the number of subcarriers. Here, each $\mathbf{h}_i[j]$ value is a complex number and denotes the channel frequency response (CFR) at subcarrier i at time j over multiple paths and is defined by

$$\mathcal{H}[t] = \sum_{i=1}^N \alpha_i(t) e^{-j2\pi f \frac{d_i(t)}{c}},$$

Symbol	Description
PM	Physical model integration stage
PL	Parameter learning stage
$\mathbf{X}$	CSI feature vector
θ	Parameters of <i>NCL</i>
T_t	Actual temperature at time t from <i>PM</i> stage
$T_{PM}(t)$	Predicted temperature at time t from <i>PM</i> stage
$\mathcal{L}_{PM}$	Loss function of physical model
θ_w^*	Estimated parameters for each window
$\hat{\theta}_i$	Predicted parameters from <i>PL</i> stage
$T_{PL}(t)$	Predicted temperature at time t from <i>PL</i> stage
$\hat{T}_{final}(t)$	Final predicted temperature at time t from the <i>PMLF</i>
$\mathcal{R}(t)$	Difference between the predictions of <i>PL</i> and <i>PM</i> : $T_{PL}(t) - T_{PM}(t)$
$\lambda_{cool}(t)$	Rate of temperature change
$\phi_{time}(t)$	Temporal features
$\mathbf{I}$	Interaction features between CSI features and physical features
W	Window size for parameter learning
$w(t)$	Cooling phase weight at time t
$\mathcal{L}_{XGB_{final}}$	Weighted loss function used for training the final XGBoost model
XGB_{param}	XGBoost model used to predict <i>NCL</i> parameters from CSI
XGB_{final}	Final XGBoost model used for temperature prediction

TABLE I: Notations and descriptions.

where N represents the number of paths, $d_i(t)$ denotes the length of the i -th path, $\alpha_i(t)$ is the complex variable that consists of the phase and amplitude attenuation information, f is the carrier frequency, and c is the speed of light.

WiFi sensing methods use a set of CSI values and apply some signal preprocessing steps to remove anomalies and smooth the data. Then, this data is used for different types of machine learning approaches [31], [32] to achieve application-specific goals.

In this study, we collect the CSI frames using ESP32 microcontrollers and the ESP32-based CSI extraction tool [3]. Out of the total 64 subcarriers, only 52 of them contain actual CSI data that is not static or zero, and thus, we use only the data of these subcarriers.

IV. PROPOSED METHOD

We propose a contact-free system that can measure the temperature of a liquid inside a container using WiFi signals. Our methodology depends on our empirical observations showing the relationship between the amplitude of the signal with the temperature of the medium the signal propagates.

A. System Overview

Fig. 3 depicts a high level overview of the proposed system, which consists of three stages. It starts with data collection. After that, we preprocess the collected data and introduce it to our proposed Physics-Guided Machine Learning Framework (*PMLF*).

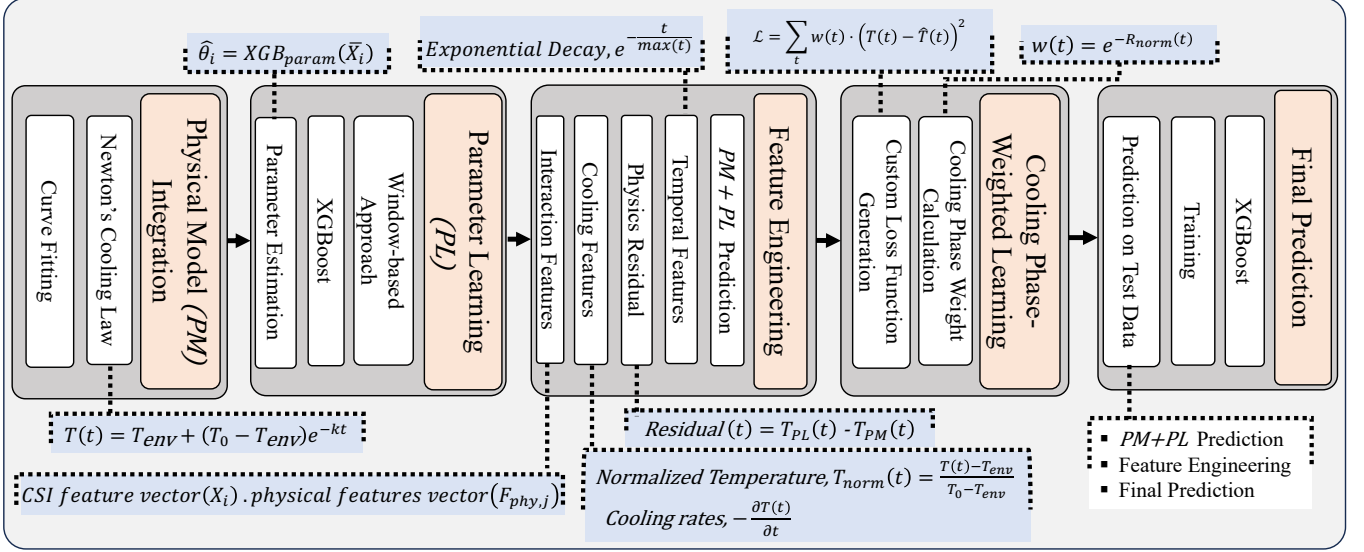


Fig. 4: Physics-Guided Machine Learning Framework (PMLF)

In this study, we use only amplitude values, as our empirical analysis showed that phase information did not offer additional insights. After denoising and normalizing the amplitudes, we apply Principal Component Analysis (PCA) to extract key subcarrier combinations most correlated with the liquid's temperature. These components are then fed into our proposed physics-guided machine learning approach.

B. Physics-Guided Machine Learning Framework (PMLF)

Fig. 4 illustrates the proposed *PMLF*, which operates in five main stages, each comprising multiple internal components. We next describe each stage in detail. Table I summarizes all notations used throughout the paper.

1) *Physical Model (PM) Integration*: Our physical model integration begins with applying *NCL* to approximate the behavior of liquid over time. This law provides a theoretical foundation for predicting how a liquid's temperature approaches the ambient temperature over time. We construct a cooling model based on this law, which will be referred as *physical model* or *PM* throughout this work. To align this physical model with experimental data, we employ a curve fitting approach to estimate the parameters of *NCL*, $\theta = \{T_0, T_{env}, k\}$, where θ represents the parameter set to be optimized. Using the temperature data (T_t), which is the actual ground truth temperature, and the corresponding time indices (t) with total time N , parameters are estimated by minimizing the squared error loss between the actual temperature and predicted temperature using,

$$\mathcal{L}_{PM} = \sum_{t=1}^N (T_t - T_{PM}(t, \theta))^2, \quad (3)$$

where T_{PM} represents the physical model based estimation of the temperature.

2) *Parameter Learning (PL)*: We build the parameter learning (*PL*) stage from the output of the physical model by leveraging machine learning to improve the model's predictive capabilities. In this stage, we employ a window-based approach to divide the data into smaller time segments, enabling localized learning of cooling behaviors. We segment the data into fixed-sized non-overlapping windows of length W . For each window, we extract the CSI features at time t (X_t , which refers to PCA components obtained from amplitude values) and the corresponding ground truth temperature values (T_t). The time indices within each window are adjusted relative to the window's start to ensure that the model captures localized cooling behaviors. We define the i^{th} window by:

$$\mathcal{W}_i = \{(X_t, T_t) \mid t \in [iW, (i+1)W)\}. \quad (4)$$

Each window consists of the CSI feature vectors at each time step and corresponding ground-truth temperature values.

We reconstruct the cooling curve for each window using the parameters predicted by the XGBoost model with the output as predicted temperature T_{PL} . For each of the windows, we take the initial temperature, T_{w0}^i , and the ending temperature of the window T_{wx}^i . This is compared to the initial and ambient temperature of *NCL*. This is like a mini *PM* stage for all the windows. Within each window, we then fit *NCL* based curve to estimate the parameters, $\theta_w^i = \{T_{w0}^i, T_{wx}^i, k^i\}$. Once the parameters are estimated for all windows, we train an XGBoost model XGB_{param} to predict these parameters directly from the mean CSI features of each window. Let $\bar{X}_i$ denote the mean CSI feature vector for window i ,

$$\bar{X}_i = \frac{1}{W} \sum_{t \in \mathcal{W}_i} X_t. \quad (5)$$

We define the parameter prediction as,

$$\hat{\theta}_i = XGB_{param}(\bar{X}_i, \theta_w^i). \quad (6)$$

where $\hat{\theta}_i = \{\hat{T}_0^i, \hat{T}_{env}^i, \hat{k}^i\}$ is the set of the predicted parameters for $\mathcal{W}_i$.

3) *Feature Engineering*: At this point, we combine outputs from the *PM* and *PL* stages to generate a rich set of features that encode both physical and data-driven insights. These predictions refine the cooling curve by incorporating CSI-based insights. We also consider temporal dynamics of the cooling process, capturing the exponential decay behavior and normalized time progression. We compute it by

$$\phi_{time}(t) = \left[e^{-\frac{t}{\max(t)}}, \frac{t}{\max(t)} \right], \quad (7)$$

where $e^{-\frac{t}{\max(t)}}$ captures the decay over time, and $\frac{t}{\max(t)}$ computes the normalized time index to reflect the relative stage of cooling. We also introduce some cooling features derived from the rate and magnitude to temperature change. The first feature we use is the rate of change of the temperature, defined and approximated as

$$\lambda_{cool}(t) = -\frac{\partial T(t)}{\partial t} \approx -\frac{T(t+1) - T(t)}{\Delta t}. \quad (8)$$

Next, we use the normalized deviation of the current temperature from the ambient temperature, which is defined by

$$T_{norm}(t) = \frac{T_t - T_{env}}{T_0 - T_{env}}. \quad (9)$$

We also employ physics residual feature, which measures the difference between the predictions of the physical model and the parameter learning model:

$$\mathcal{R}(t) = T_{PL}(t) - T_{PM}(t). \quad (10)$$

This feature highlights regions where the parameter learning model corrects the physical model, emphasizing deviations caused by environmental or experimental variations. To investigate the interplay between CSI features and physical insights, we also adopt interaction features. For a given CSI feature vector $\mathbf{X}$ and the comprehensive physical feature vector $\mathbf{F}_{phy}$, we compute pairwise interactions as:

$$\mathbf{I}(i, j) = X_i \cdot F_{phy, j}, \quad (11)$$

where X_i and $F_{phy, j}$ are the i -th and j -th components of $\mathbf{X}$ and $\mathbf{F}_{phy}$, respectively. The physical feature vector $\mathbf{F}_{phy}$ is constructed as:

$$\mathbf{F}_{phy} = [T_{PM}(t), T_{PL}(t), \lambda_{cool}(t), T_{norm}(t), \mathcal{R}(t), \phi_{time}(t)] \quad (12)$$

These interaction features are then defined as:

$$\mathbf{I} = \{X_i F_{phy, j} \mid i \in [1, n_X], j \in [1, n_{phy}]\}, \quad (13)$$

where n_X and n_{phy} represent the dimensions of the CSI feature vector $\mathbf{X}$ and the physical feature vector $\mathbf{F}_{phy}$, respectively.

Finally, the resulting feature set is formed by combining all the above features,

$$\mathbf{F}(t) = [\mathbf{F}_{phy}, \mathbf{I}(t)]. \quad (14)$$

This consolidated feature set ensures that the model has access to a comprehensive representation of the cooling dynamics from *NCL* as well as the CSI fingerprints from the signal propagation.

4) *Cooling Phase-Weighted Learning*: In this process, we construct a weighted loss function to enhance the training process by incorporating phase-specific weights to prioritize different cooling phases. These weights are derived from the cooling rate, highlighting quick cooling stages where temperature fluctuations are more significant.

In early stages of cooling, the temperature difference between the liquid and the environment is larger. According to *NCL*, this results in faster cooling rate, $\lambda_{cool}(t)$. When the temperature of the liquid approaches ambient temperature, $\lambda_{cool}(t)$ slows down. Thus, faster cooling is responsible for a higher absolute value of $\lambda_{cool}(t)$. By normalizing this value, we scale the values to a consistent range. Then we customize the weight values $w(t)$, using an exponential function $e^{\lambda_{norm}(t)}$. This ensures that higher cooling rates are assigned a higher weight and slower rates are assigned smaller weights.

By introducing these weights during training, we ensure the model effectively adjusts to both rapid and gradual cooling behaviors. We calculate the cooling phase weights based on the normalized cooling rate of the temperature. In practice, we approximate this rate as described in Eq. 8. Then we normalize the cooling rate, denoted by λ_{norm} , and compute the weights from it as,

$$w(t) = e^{-\lambda_{norm}(t)}. \quad (15)$$

This weight is used in the next stage when we train the final XGBoost model. Using the engineered feature set at time t , i.e., $\mathbf{F}(t)$, and the predicted and actual temperatures, $\hat{T}(t)$ and $T(t)$, respectively, we define a mean squared error (MSE) based weighted loss function as follows,

$$\mathcal{L}_{XGB_{final}} = \sum_t w(t) \left(T(t) - \hat{T}(t) \right)^2, \quad (16)$$

where $w(t)$ modulates the contribution of each time step t to the total loss. By using this weighted loss function, the model learns to balance the importance of rapid and slow cooling phases.

5) *Final Prediction*: This is the final stage of our proposed *PMLF*. In this stage, the insights from the feature engineering process are used to deliver temperature predictions. Here, we use a gradient boosting regressor (XGBoost) as our final predictive model. The model is trained on the comprehensive feature set generated in the Feature Engineering stage and evaluated on unseen test data.

For the final prediction, we pass the dataset through the *PM* and *PL* stages of *PMLF*. With the predictions of *PM*

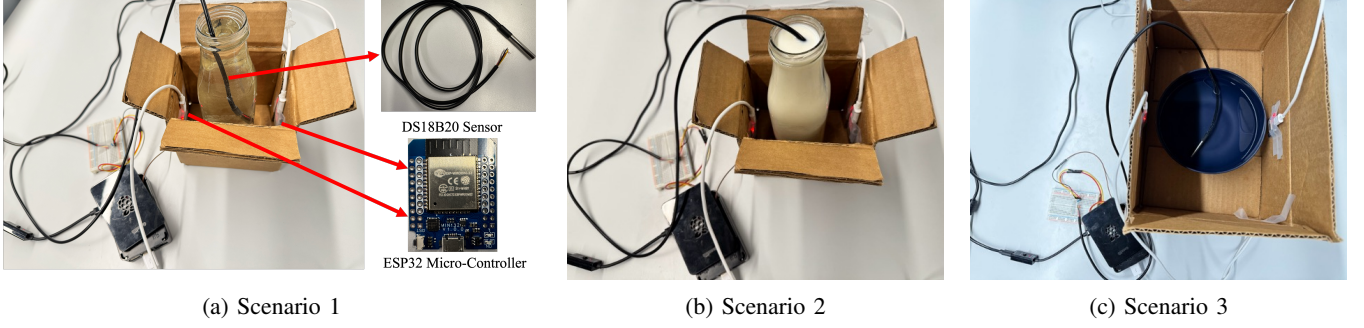


Fig. 5: Experimental setups for different scenarios (a) Water in a glass container, (b) Milk in a glass container, (c) Water in a larger plastic container.

Component	Specification	Purpose
ESP32	WiFi-enabled Micro-controller	TX and RX for CSI collection
DS18B20	Digital temperature sensor	Ground truth temperature measurement
Raspberry Pi	Model 4B with 2GB RAM	Time synchronization and data management

TABLE II: Hardware used during experiments.

and PL , the test data undergoes the same feature engineering steps as the training data. The trained XGBoost model takes the engineered features $\mathbf{F}_{test}(t)$ as input to predict the final temperature values:

$$\hat{T}_{final}(t) = XGB_{final}(\mathbf{F}_{test}(t)), \quad (17)$$

where XGB_{final} represents the trained XGBoost regressor.

V. EVALUATION

In this section, we first describe our experiments and discuss feature analysis results. After that we present the evaluation results of the proposed approach.

A. Experimental Setup and Data Collection

We designed the experiments to collect WiFi CSI in a controlled environment, with focus on sensing the temperature of liquid through signal variations. The setup consists of two ESP32 microcontrollers, a DS18B20 liquid temperature sensor [33], and a Raspberry Pi.

We collected data for three different scenarios, as shown in Fig. 5.

- *Scenario 1*: Hot water in a glass bottle which is the primary data collected for testing the system.
- *Scenario 2*: Hot milk in a glass bottle which is used to introduce liquid variation.
- *Scenario 3*: Hot water in a plastic bowl which is larger than glass bottle. This case is considered to introduce liquid container variation.

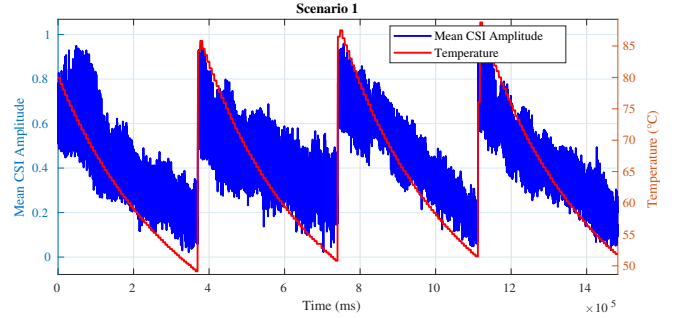


Fig. 6: Temporal evolution of mean CSI amplitude and water temperature during the cooling process in Scenario 1.

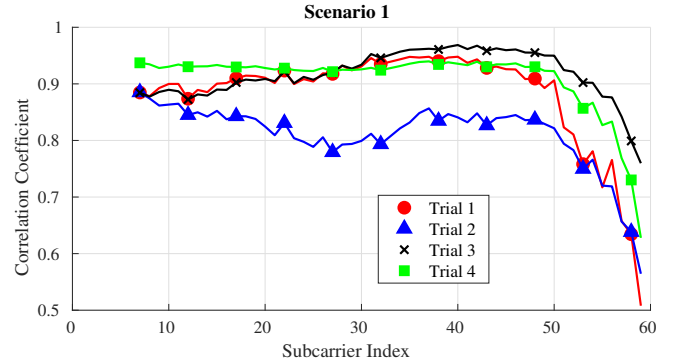


Fig. 7: Correlation between CSI amplitude and temperature for each subcarrier across four experimental trials in Scenario 1.

The hardware used during experimental setups is summarized in Table II. We used the ESP32-CSI-Toolkit [3], [34] to collect CSI using a pair of ESP32 WiFi-enabled microcontrollers as a TX and RX, respectively. The DS18B20 sensor measured the liquid's temperature during cooling. The selection of channel can be crucial in data collection [35]. Here, the ESP32 devices captured CSI data from WiFi signals transmitted over channel 1 at a packet rate of 100Hz.

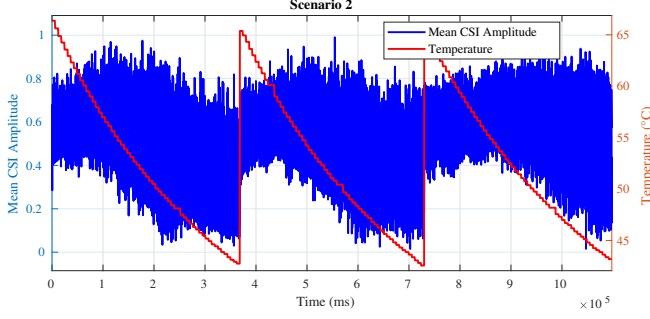


Fig. 8: Temporal evolution of mean CSI amplitude and milk temperature during the cooling process in Scenario 2.

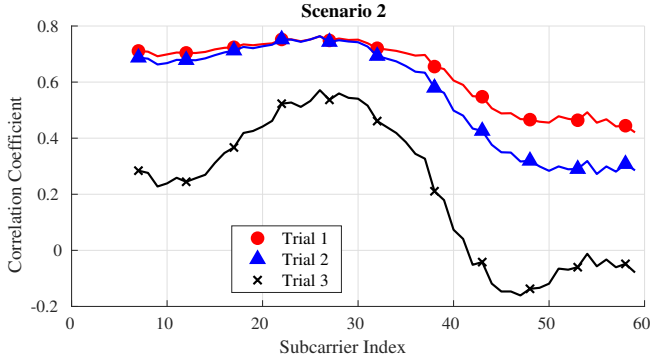


Fig. 9: Correlation between CSI amplitude and temperature for each subcarrier across three experimental trials in Scenario 2.

The ESP32 devices and the DS18B20 sensor are connected to a Raspberry Pi to ensure precise time synchronization and continuous logging of both CSI and temperature data. Note that, the liquid container is in the Line-of-Sight (LOS) of the TX-RX pair.

For the first scenario, we conducted four trials, each one of them lasting for one hour. At the start of each trial, the water was heated to approximately 80°C and allowed to cool naturally for one hour. During this cooling process, the DS18B20 sensor recorded the liquid temperature at one-minute intervals for the ground truth. The ESP32 devices captured CSI frames continuously, focusing on signal variations across 64 subcarriers. As for the second and third scenarios, data was collected in an identical fashion as before for three trials.

B. Feature Analysis and Selection

The relationship between the mean CSI amplitude over all subcarriers and water temperature during the cooling process in the first scenario is shown in Fig. 6. The red curve represents the ground-truth temperature recorded from the DS18B20 sensor, while the blue curve depicts the mean CSI

amplitude. The mean amplitude is computed by averaging the amplitudes across all 52 active subcarriers for each CSI frame, providing a single aggregated feature that captures signal dynamics over time. The results highlight a strong correlation between the mean CSI amplitude and the cooling dynamics of the liquid. As the water cools naturally from approximately 85°C to 50°C, the mean CSI amplitude consistently decreases, mirroring the exponential decay of the temperature. This relationship suggests that variations in the WiFi CSI data are closely tied to the physical changes in the liquid's properties, such as refractive index, during cooling.

Fig. 7 shows the correlation coefficient between CSI amplitude and the temperature for each subcarrier across all four trials. The results show a high correlation for most of the subcarriers, with even the lowest coefficient being above 0.5, and only a few subcarriers are showing differences from the rest of them. This suggests that we can consider CSI amplitude from all the subcarriers for the proposed liquid temperature sensing system.

For the second scenario, the mean CSI amplitude also aligns with the liquid temperature as shown in Fig. 8, but the CSI amplitude exhibits higher noise and irregularities compared to the water dataset in the first scenario. This may be due to the higher viscosity and density of milk, which can introduce additional scattering effects or changes in signal propagation. Milk's higher refractive index and absorption properties (due to its composition of fats, proteins, and suspended particles) [36] could contribute to greater signal attenuation and scattering [37], resulting in the observed differences in CSI amplitude behavior. Fig. 9 demonstrates the correlation coefficient of amplitude for each of the active subcarriers. While some subcarriers are not well correlated, there is still a good subset of subcarriers (e.g., 20-30) that can be leveraged for accurate predictions. We expect that PCA can help identify these highly correlated subcarrier data within the top PCA components.

In the dataset collected with a larger container, the larger exposed surface area accelerates heat transfer via convection [38] and evaporation [39]. These may cause potential issues during data collection; thus, in this scenario, we used data from the earlier stages of this experimental setup. The mean amplitude values as well as correlation results are similar to the second scenario, thus we did not add the corresponding figures.

C. Evaluation Metrics

To evaluate the performance of the proposed system, we compute commonly used regression metrics. Let $T_{test}(t)$ represent the actual temperature at time t , $\hat{T}_{final}(t)$ represent the predicted temperature at time t , N_s denote the total number of test samples, and $\bar{T}_{test}$ denote the mean of the observed temperatures. Then, the evaluation metrics we use are computed as:

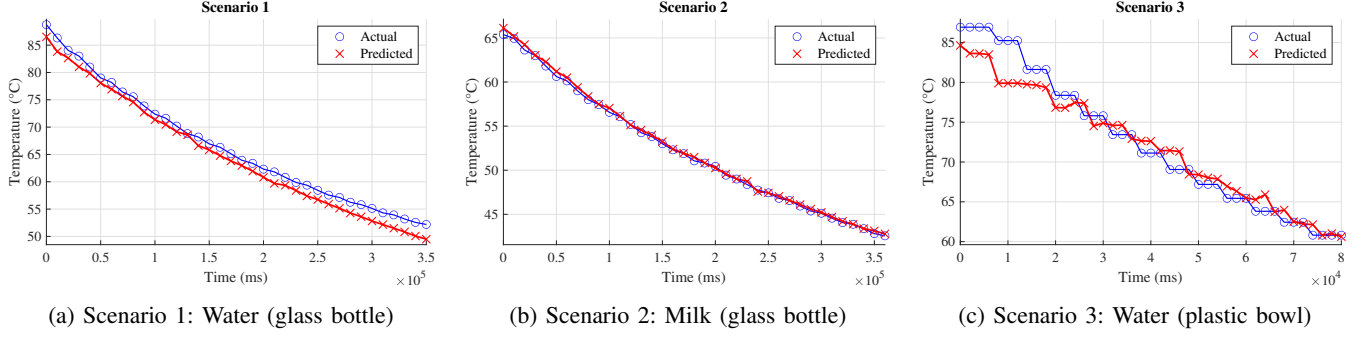


Fig. 10: Actual and predicted temperature for *PMLF* in different scenarios.

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N_s} \sum_{t=1}^{N_s} |T_{test}(t) - \hat{T}_{final}(t)|. \quad (18)$$

The MAE score measures the average absolute difference between predicted and observed temperatures.

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N_s} \sum_{t=1}^{N_s} (T_{test}(t) - \hat{T}_{final}(t))^2}. \quad (19)$$

The RMSE score penalizes larger errors more heavily.

- R^2 Score:

$$R^2 = 1 - \frac{\sum_{t=1}^{N_s} (T_{test}(t) - \hat{T}_{final}(t))^2}{\sum_{t=1}^{N_s} (T_{test}(t) - \bar{T}_{test})^2}. \quad (20)$$

The R^2 score quantifies the proportion of variance explained by the model.

D. Results

In this section, we discuss the results of our experimental evaluation. For comparative evaluation, we use the XGBoost regression model to compare with *PMLF*. The model is configured with a squared error objective function and trained using a learning rate of 0.05, a maximum tree depth of 6, and 1400 estimators, with subsampling and column sampling rates set to 0.7. We also employ two XGBoost models in *PMLF*. The first model, XGB_{param} , is trained to predict the parameters of Newton's Cooling Law from the mean CSI features within each window, enhancing the parameter learning stage. The second model, XGB_{final} , serves as the final prediction model, utilizing the engineered features from the feature engineering stage to predict the temperature. Both models use a learning rate of 0.01, a maximum tree depth of 6, and 2000 estimators. These parameters are obtained through an empirical search and trial based optimization for both approaches. Table III presents the summary of performance comparison of *PMLF* and the baseline XGBoost model across the three experimented scenarios.

Scenario	Model	MAE	RMSE	R^2 Score
1	XGBoost	3.55	4.37	0.83
	<i>PMLF</i>	1.59	1.69	0.97
2	XGBoost	1.75	2.18	0.89
	<i>PMLF</i>	0.22	0.27	0.99
3	XGBoost	2.95	4.18	0.75
	<i>PMLF</i>	1.58	2.07	0.94

TABLE III: Prediction results for different scenarios.

1) *Scenario 1*: In the first scenario with water in a glass bottle, *PMLF* shows lower MAE and RMSE of 1.59°C and 1.69°C, respectively, compared to the XGBoost model that provides an MAE of 3.55°C and an RMSE of 4.37°C. *PMLF* also shows great R^2 score compared to XGBoost model. Fig. 10a depicts the *PMLF* predictions compared to the ground truth values, which shows a close alignment.

2) *Scenario 2*: With the milk dataset collected in a glass bottle, despite more variations and less alignment of amplitude values with temperature (Fig. 8), we obtained surprisingly better results. In this case, *PMLF* outperforms XGBoost with an MAE of 0.22°C and an RMSE of 0.27°C, achieving a near-perfect R^2 score of 0.99. This indicates that nearly all variation in the temperature data is captured by the model. Predictions for this case are given in Fig. 10b.

3) *Scenario 3*: In the last scenario, where the cooling dynamics are more complex due to a larger surface area and faster initial cooling, *PMLF* still shows considerable improvement over XGBoost with the available data. It achieves RMSE of 2.07°C compared to 4.18°C of XGBoost, and an R^2 score of 0.94 against 0.75. Fig. 10c shows the predictions for this scenario, where we observe more variation in results.

VI. CONCLUSION

In this work, we have explored the liquid temperature measurement problem using fine-grained WiFi signals. We have utilized amplitude data from the CSI values over multiple subcarriers together with the principles defined in Newton's Law of Cooling to develop a physics-guided machine learning solution. Through experiments performed with two different liquids (i.e., water, milk) and two different containers (i.e., glass, plastic) with different sizes, we have shown promising results that can achieve 0.22-1.59 MAE. These results are better than the machine learning only

approach, showing the benefit of the proposed novel physics-guided machine learning approach. In our future work, we will be exploring new liquids and containers, as well as consider longer distances between TX-RX devices.

VII. ACKNOWLEDGMENTS

High Performance Computing resources provided by the High Performance Research Computing (HPRC) core facility at Virginia Commonwealth University (<https://hprc.vcu.edu>) were used for conducting the research reported in this work.

REFERENCES

- [1] Linux 802.11n CSI Tool, 2019. [Online]. Available: <https://dhalperi.github.io/linux-80211n-csitol/>.
- [2] Nexmon: The c-based firmware patching framework, 2019. [Online]. Available: <https://nexmon.org>.
- [3] S. M. Hernandez, "ESP32 CSI Tool," 2023. [Online]. Available: <https://stevenmhernandez.github.io/ESP32-CSI-Tool/>
- [4] S. M. Hernandez and E. Bulut, "Performing WiFi Sensing with Off-the-shelf Smartphones," in *IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–3.
- [5] Y. Zeng, P. H. Pathak, and P. Mohapatra, "WiWho: WiFi-based person identification in smart spaces," in *15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2016, pp. 1–12.
- [6] D. Wu, Y. Zeng, R. Gao, S. Li, Y. Li, R. C. Shah, H. Lu, and D. Zhang, "WiTraj: Robust indoor motion tracking with WiFi signals," *IEEE Trans. on Mobile Computing*, vol. 22, no. 5, pp. 3062–3078, 2021.
- [7] Y. Lin, Y. Gao, B. Li, and W. Dong, "Revisiting indoor intrusion detection with wifi signals: Do not panic over a pet!" *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10437–10449, 2020.
- [8] C. Feng, J. Xiong, L. Chang, J. Wang, X. Chen, D. Fang, and Z. Tang, "WiMi: Target material identification with commodity Wi-Fi devices," in *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 700–710.
- [9] Y. Ren, S. Tan, L. Zhang, Z. Wang, Z. Wang, and J. Yang, "Liquid Level Sensing Using Commodity WiFi in a Smart Home Environment," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, pp. 1–30, 2020.
- [10] S. Zhang, R. H. Venkatnarayan, and M. Shahzad, "A wifi-based home security system," in *IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 129–137.
- [11] J. Y. Eom, S. U. Jang, and W. S. Jeon, "Wi-Sniffer: Wifi-based intruder detection system using deep learning and decision tree," in *IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–7.
- [12] M. McDonough, T. Moomaw, M. Touhiduzzaman, and E. Bulut, "WiAlert: WiFi Sensing for Real-time Package Theft Alerts at Residential Doorsteps," in *Proceedings of the ACM 24th MobiHoc - The 8th National Workshop for REU Research in Networking and Systems (REUNS), October 23-26, 2023, Washington, DC, USA.*, pp. 424–429.
- [13] C. Abram, B. Fond, and F. Beyrau, "Temperature measurement techniques for gas and liquid flows using thermographic phosphor tracer particles," *Progress in energy and combustion science*, vol. 64, pp. 93–156, 2018.
- [14] F. Zhang, K. Niu, X. Fu, and B. Jin, "AcousticThermo: Temperature monitoring using acoustic pulse signal," in *16th International Conference on Mobility, Sensing and Networking (MSN)*, 2020, pp. 683–687.
- [15] C. Cai, Z. Chen, H. Pu, L. Ye, M. Hu, and J. Luo, "AcuTe: Acoustic thermometer empowered by a single smartphone," in *Proc. of the 18th conference on embedded networked sensor systems*, 2020, pp. 28–41.
- [16] X. Chen, J. Liu, F. Xiao, S. Chen, and L. Chen, "Thermotag: Item-level temperature sensing with a passive rfid tag," in *Proc. of the 19th annual international conference on mobile systems, applications, and services*, 2021, pp. 163–174.
- [17] S. Pradhan and L. Qiu, "Rtsense: Passive rfid based temperature sensing," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, 2020, pp. 42–55.
- [18] S. Depatla and Y. Mostofi, "Crowd counting through walls using WiFi," in *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018, pp. 1–10.
- [19] S. M. Hernandez and E. Bulut, "Adversarial occupancy monitoring using one-sided through-wall WiFi sensing," in *IEEE International Conference on Communications (ICC)*, 2021, pp. 1–6.
- [20] Y. Liu, L. Jiang, L. Kong, Q. Xiang, X. Liu, and G. Chen, "Wi-Fruit: See through fruits with smart devices," *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 4, pp. 1–29, 2021.
- [21] M. Touhiduzzaman, S. M. Hernandez, P. E. Pidcoe, and E. Bulut, "Wi-PT-Hand: Wireless Sensing based Low-cost Physical Rehabilitation Tracking for Hand Movements," *ACM Trans. Comput. Healthcare*, vol. 6, no. 1, Jan. 2025.
- [22] W. Akpabio and E. Bulut, "PhysiFi: WiFi Sensing for Monitoring Therapeutic Robotic Systems," in *IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, 2025.
- [23] Y. Deng, D. Mishra, S. Atakaramians, and A. Seneviratne, "Smart CSI Processing For Accurate Commodity WiFi-based Humidity Sensing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, 2024, pp. 279–283.
- [24] Y. Deng, J. Xie, C. Liu, X. Men, D. Mishra, and A. Seneviratne, "WiFi sensing based textile wetness monitoring," in *IEEE International Conference on Communications (ICC)*, 2024, pp. 4185–4190.
- [25] S. M. Hernandez, D. Erdag, and E. Bulut, "Towards Dense and Scalable Soil Sensing Through Low-Cost WiFi Sensing Networks," in *Proc. of 46th IEEE Conference on Local Computer Networks (LCN)*, Edmonton, AB, Canada, October 4-7, 2021, pp. 549–556.
- [26] J. Li, A. Sharma, D. Mishra, and A. Seneviratne, "Thermal profiling by WiFi sensing in IoT networks," in *IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.
- [27] W. Yang, X. Wang, A. Song, and S. Mao, "Wi-Wheat: Contact-free wheat moisture detection with commodity WiFi," in *IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [28] S. Zhong, Y. Huang, R. Ruby, L. Wang, Y.-X. Qiu, and K. Wu, "Wi-Fire: Device-free fire detection using WiFi networks," in *IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [29] F. Shang, P. Yang, D. Yan, S. Zhang, and X.-Y. Li, "LiquImager: Fine-grained Liquid Identification and Container Imaging System with COTS WiFi Devices," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 8, no. 1, pp. 1–29, 2024.
- [30] F. Incropera, *Fundamentals of Heat and Mass Transfer*. Wiley, 2007.
- [31] N. Fahad, M. Touhiduzzaman, and E. Bulut, "Ensemble Learning based WiFi Sensing using Spatially Distributed TX-RX Links," in *IEEE International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, Hawaii, USA, 2025, pp. 580–585.
- [32] S. Samiul Alam, A. Chakma, A. Imran, and Y. Min Jang, "DCSiNet: Effective Handling of Noisy CSI in FDD Massive MIMO System," *IEEE Wireless Commun. Letters*, vol. 13, no. 9, pp. 2556–2560, 2024.
- [33] <https://gikfun.com/products/gikfun-ds18b20-temperature-sensor-waterproof-digital-thermal-probe-sensor-for-arduino-pack-of-5pcs>, (Accessed on 12/10/2024).
- [34] S. M. Hernandez and E. Bulut, "Lightweight and Standalone IoT based WiFi Sensing for Active Repositioning and Mobility," in *21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Cork, Ireland, Jun. 2020.
- [35] N. Fahad and E. Bulut, "Channel Matters: Exploring LoS/NLoS Channel Effects on WiFi Sensing Performance," in *IEEE Conference on Computer Communications (INFOCOM) Workshops*, 2025.
- [36] S. Stocker, F. Foschum, P. Krauter, F. Bergmann, A. Hohmann, C. Scalfi Happ, and A. Kienle, "Broadband optical properties of milk," *Applied Spectroscopy*, vol. 71, no. 5, p. 951–962, Oct. 2016.
- [37] Y. Luo, L. Dai, X. Jiang, and X. Fu, "Measurement of absorption and scattering properties of milk using a hyperspectral spatial frequency domain imaging system," *Journal of Food Measurement and Characterization*, vol. 16, no. 1, p. 753–761, Oct. 2021.
- [38] J. P. Holman, *Heat Transfer*, 10th ed. McGraw-Hill Education, 2010.
- [39] T. Poós and E. Varjú, "Review for prediction of evaporation rate at natural convection," *Heat and Mass Transfer*, vol. 55, no. 6, p. 1651–1660, Dec. 2018.